Dense 3D Face Decoding over 2500FPS: Joint Texture & Shape Convolutional Mesh Decoders

Yuxiang Zhou * [1] Jiankang Deng * [1,3] Irene Kotsia[2] Stefanos Zafeiriou[1,3]
[1]Imperial College London [2]University of Middlesex [3]FaceSoft
{yuxiang.zhou10, j.deng16, s.zafeiriou}@imperial.ac.uk , i.kotsia@mdx.ac.uk

Abstract

*3D Morphable Models (3DMMs) are statistical models that represent facial texture and shape variations using a set of linear bases and more particular Principal Component Analysis (PCA). 3DMMs were used as statistical priors for reconstructing 3D faces from images by solving non-linear least square optimization problems. Recently, 3DMMs were used as generative models for training non-linear mappings (*i.e.*, regressors) from image to the parameters of the models via Deep Convolutional Neural Networks (DCNNs). Nevertheless, all of the above methods use either fully connected layers or 2D convolutions on parametric unwrapped UV spaces leading to large networks with many parameters. In this paper, we present the first, to the best of our knowledge, non-linear 3DMMs by learning joint texture and shape auto-encoders using direct mesh convolutions. We demonstrate how these auto-encoders can be used to train very light-weight models that perform Coloured Mesh Decoding (CMD) in-the-wild at a speed of over 2500 FPS.*

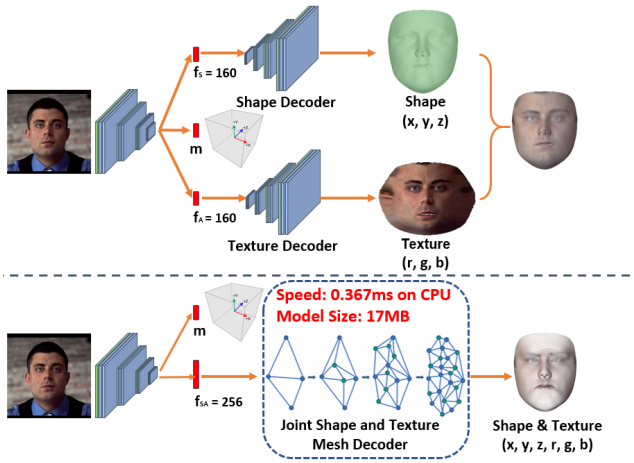


Figure 1. A typical non-linear 3DMM [38] is a DCNN trained to recover shape and texture separately when given one or more 2D images. We propose a non-linear 3DMM to jointly model shape and texture by geometric convolutional networks. Our coloured mesh decoder can run over 2500 FPS with compact model size, thus being significantly faster and smaller (in terms of parameters) when compared to the PCA model.

1. Introduction

Twenty years ago, Blanz and Vetter demonstrated a remarkable achievement [2]. They showed that it is possible to reconstruct 3D facial geometry from a single image. This was possible by solving a non-linear optimization problem whose solution space was confined by a linear statistical model of the 3D facial shape and texture, the so-called 3D Morphable Model (3DMM). Methods based on 3DMMs are still among the state-of-the-art for 3D face reconstruction, even from images captured in-the-wild [6, 4, 5].

During the past two years, a lot of works have been conducted on how to harness the power of Deep Convolutional Neural Networks (DCNNs) for 3D shape and texture estimation from 2D facial images. The first such methods either trained regression DCNNs from image to the parameters of a 3DMM [36] or used a 3DMM to synthesize images and formulate an image-to-image translation problem in order to estimate the depth, using DCNNs [31]. The recent, more sophisticated, DCNN-based methods were trained using self-supervised techniques [17, 37, 38] and made use of differentiable image formation architectures and differentiable renderers [17]. The most recent methods such as [37, 38] and [34] used self-supervision to go beyond the standard 3DMMs in terms of texture and shape. In particular, [34] used both the 3DMMs model, as well as additional network structures (called correctives) that can capture information outside the space of 3DMMs, in order to represent the shape and texture. The method in [37, 38] tried to learn non-linear spaces (*i.e.*, decoders, which are called non-linear 3DMMs) of shape and texture directly from the data. Nevertheless, in order to avoid poor training performance, these methods used 3DMMs fittings for the model pre-training.

*Equal contributions.

In all the above methods the 3DMMs, linear or non-linear in a form of a decoder, were modelled with either fully connected nodes [36] or, especially in the texture space, with 2D convolutions on unwrapped UV space [37, 38]. In this paper, we take a radically different direction. That is, motivated by the line of research on Geometric Deep Learning (GDL), a field that attempts to generalize DCNNs to non-Euclidean domains such as graphs/manifolds/meshes [33, 12, 21, 7, 27], we make the first attempt to develop a non-linear 3DMM, that describes both shape and texture, by using mesh convolutions. Apart from being more intuitive defining non-linear 3DMMs using mesh convolutions, their major advantage is that they are defined by networks that have a very small number of parameters and hence can have very small computational complexity. In summary, the contributions of our paper are the following:

- We demonstrate how recent techniques that find dense or sparse correspondences (*e.g.*, densereg [18], landmark localization methods [40]) can be easily extended to estimate 3D facial geometric information by means of mesh convolutional decoders.
- We present the first, to the best of our knowledge, non-linear 3DMM using mesh convolutions. The proposed method decodes both shape and texture directly on the mesh domain with a compact model size (17MB) and amazing efficiency (over 2500 FPS on CPU). This decoder is different to the recently proposed decoder in [27] which only decodes 3D shape information.
- We propose an encoder-decoder structure that reconstructs the texture and shape directly from an in-the-wild 2D facial image. Due to the efficiency of the proposed Coloured Mesh Decoder (CMD), our method can estimate the 3D shape over 300 FPS (for the entire system).

2. Related Work

In the following, we briefly touch upon related topics in the literature such as linear and non-linear 3DMM representations.

Linear 3D Morphable Models. For the past two decades, the method of choice for representing and generating 3D faces was Principal Component Analysis (PCA). PCA was used for building statistical 3D shape models (*i.e.*, 3D Morphable Models (3DMMs)) in many works [2, 3, 29]. Recently, PCA was adopted for building large-scale statistical models of the 3D face [6] and head [11]. It is very convenient for representing and generating faces to decouple facial identity variations from expression variations. Hence, statistical blend shape models were introduced representing only the expression variations using PCA [22, 9]. The original 3DMM [2] used a PCA model for also describing the texture variations. Nevertheless, this is quite limited in describing the texture variability in image captured in-the-wild conditions.

Non-linear 3D Morphable Models. In the past year, the first attempts for learning non-linear 3DMMs were introduced [37, 38, 34]. These 3DMMs can be regarded as decoders that use DCNNs, coupled with an image-encoder. In particular, the method [34] used self-supervision to learn a new decoder with fully-connected layers that combined a linear 3DMM with new structures that can reconstruct arbitrary images. Similarly, the methods [37, 38] used either fully connected layers or 2D convolutions on a UV map for decoding the shape and texture.

All the above methods used either fully connected layers or 2D convolutions on unwrapped spaces to define the non-linear 3DMM decoders. However, these methods lead to deep networks with a large number of parameters and do not exploit the local geometry of the 3D facial structure. Therefore, decoders that use convolutions directly in the non-Euclidean facial mesh domain should be built. The field of deep learning on non-Euclidean domains, also referred to as Geometric Deep Learning [7], has recently gained some popularity. The first works included [23] that proposed the so-called MeshVAE which trains a Variational-Auto-Encoder (VAE) using convolutional operators from [39] and CoMA [27] that used a similar architecture with spectral Chebyshev filters [12] and additional spatial pooling to generate 3D facial meshes. The authors demonstrated that CoMA can represent better faces with expressions than PCA in a very small dimensional latent space of only eight dimensions.

In this paper, we propose the first auto-encoder that directly uses mesh convolutions for joint texture and shape representation. This brings forth a highly effective and efficient coloured mesh decoder which can be used for 3D face reconstruction for in-the-wild data.

3. Proposed Approach

3.1. Coloured Mesh Auto-Encoder

Mesh Convolution. We define our mesh auto-encoder based on the un-directed and connected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \in \mathbb{R}^{n\times 6}$ is a set of n vertices containing the joint shape (*e.g.* x, y, z) and texture (*e.g.* r, g, b) information, and $\mathcal{E} \in \{0,1\}^{n\times n}$ is an adjacency matrix encoding the connection status between vertices.

Following [12, 26], the non-normalized graph Laplacian is defined as $L = D - \mathcal{E} \in \mathbb{R}^{n\times n}$ where $D \in \mathbb{R}^{n\times n}$ is the diagonal matrix with $D_{ii} = \sum_j \mathcal{E}_{ij}$ and the normalized definition is $L = I_n - D^{-1/2}\mathcal{E}D^{-1/2}$ where I_n is the identity matrix. The Laplacian L can be diagonalized by the Fourier bases $U = [u_0, \ldots, u_{n-1}] \in \mathbb{R}^{n\times n}$ such that $L = U\Lambda U^T$ where $\Lambda = \mathrm{diag}([\lambda_0, \ldots, \lambda_{n-1}]) \in \mathbb{R}^{n\times n}$. The graph

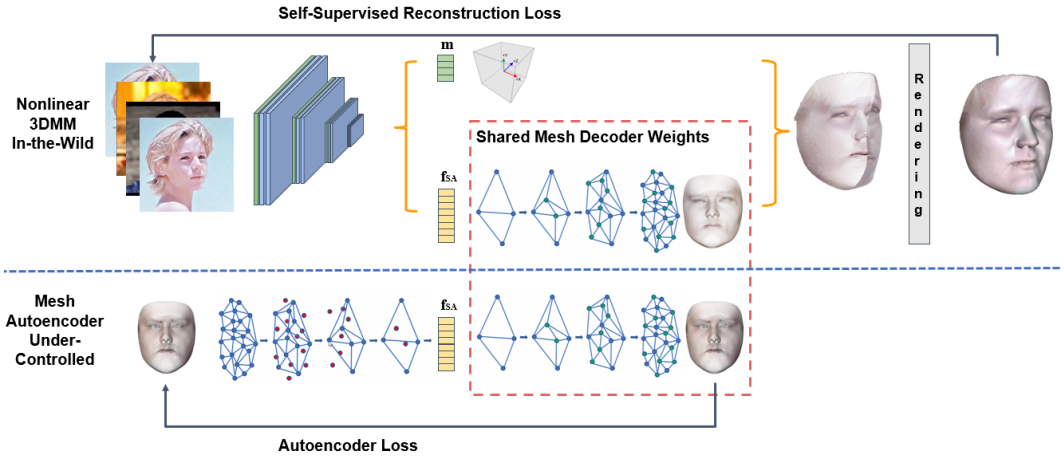


Figure 2. Training procedure of the proposed method. For controlled data, we employ auto-encoder loss. For in-the-wild data, we exploit self-supervised reconstruction loss. Both models are trained end-to-end jointly with a shared coloured mesh decoder.

Fourier transform of our face representation $x \in \mathbb{R}^{n\times 6}$ is then defined as $\hat{x} = U^T x$, and its inverse as $x = U\hat{x}$.

The operation of the convolution on a graph can be defined by formulating mesh filtering with a kernel g_θ using a recursive Chebyshev polynomial [12, 26]. The filter g_θ can be parameterized as a truncated Chebyshev polynomial expansion of order K,

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}), \tag{1}$$

where $\theta \in \mathbb{R}^K$ is a vector of Chebyshev coefficients and $T_k(\tilde{\Lambda}) \in \mathbb{R}^{n\times n}$ is the Chebyshev polynomial of order k evaluated at a scaled Laplacian $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_n$. T_k can be recursively computed by $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$.

The spectral convolution can be defined as

$$y_j = \sum_{i=1}^{F_{in}} g_{\theta_{i,j}}(L)x_i, \tag{2}$$

where $x \in \mathbb{R}^{n\times F_{in}}$ is the input and $y \in \mathbb{R}^{n\times F_{out}}$ is the output. The entire filtering operation $y = g_\theta(L)x$ is very efficient and only costs $\mathcal{O}(K|\mathcal{E}|)$ operations.

Mesh Down-sampling and Up-sampling. We follow [26] to employ a binary transformation matrix $Q_d \in \{0,1\}^{n\times m}$ to perform down-sampling of a mesh with m vertices and conduct up-sampling using another transformation matrix $Q_u \in \mathbb{R}^{m\times n}$.

Q_d is calculated by iteratively contracting vertex pairs under the constraint of minimizing quadric error [15]. During down-sampling, we store the barycentric coordinates of the discarded vertices with regard to the down-sampled mesh so that the up-sampling step can add new vertices with the same barycentric locations information.

For up-sampling, vertices directly retained during the down-sampling step undergo convolutional transformations. Vertices discarded during down-sampling are mapped into the down-sampled mesh surface using recorded barycentric coordinates. The up-sampled mesh with vertices $\mathcal{V}_u$ is efficiently predicted by a sparse matrix multiplication, $\mathcal{V}_u = Q_u\mathcal{V}_d$.

3.2. Coloured Mesh Decoder in-the-Wild

The non-linear 3DMM fitting in-the-wild is designed in an unsupervised/self-supervised manner. As we are able to construct joint shape & texture bases with the coloured mesh auto-encoder, the problem can be treated as a matrix multiplication between the bases and the optimal coefficients that reconstruct the 3D face. From the perspective of a neural network, this can be viewed as an image encoder $E_I(I;\theta_\mathbf{I})$ that is trained to regress to the 3D shape and texture, noted as f_{SA}. As shown in Fig. 2, a 2D convolution network is used to encode in-the-wild images followed by a mesh decoder $\mathcal{D}(\mathbf{f_{SA}};\theta_\mathbf{D})$, whose weights are shared across the decoder [10] in the mesh auto-encoder. However, the output of the joint shape & texture decoder is a coloured mesh within a unit sphere. Like linear 3DMM [4], a camera model is required to project the 3D mesh from the object-centered Cartesian coordinates into an image plane in the same Cartesian coordinates.

Projection Model. We employ a pinhole camera model in this work, which utilizes a perspective transformation model. The parameters of the projection operation can be formulated as following:

$$\mathbf{c} = [p_x, p_y, p_z, o_x, o_y, o_z, u_x, u_y, u_z, f]^\mathsf{T}, \tag{3}$$

where $\mathbf{p}, \mathbf{o}, \mathbf{u}$ represent camera position, orientation and up-right direction, respectively, in Cartesian coordinates. f is the field of view (FOV) that controls the perspective projec-

tion. We also concatenate lighting parameters together with camera parameters as rendering parameters that will be predicted by the image encoder. Three point light sources and constant ambient light are assumed, to a total of 12 parameters $\mathbf{l}$ for lighting. For abbreviation, we represent the rendering parameter $\mathbf{m} = [\mathbf{c}^T, \mathbf{l}^T]^T$ as a vector of size 22 and the projection model as the function $\hat{\mathbf{I}} = \mathcal{P}(\mathcal{D}(\mathbf{f_{SA}}); \mathbf{m}) : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{2N}$.

Differentiable Renderer. To make the network end-to-end trainable, we incorporated a differentiable renderer [17] to project the output mesh $\mathcal{D}(\mathbf{f_{SA}})$ onto the image plane $\hat{\mathbf{I}}$. The l_1 norm is pixel-wisely calculated as the loss function. The renderer, also known as rasterizer, generates barycentric coordinates and corresponding triangle IDs for each pixel at the image plane. The rendering procedure involves Phong shading [25] and interpolating according to the barycentric coordinates. Also, camera and illumination parameters are computed in the same framework. The whole pipeline is able to be trained end-to-end with the loss gradients back-propagated through the differentiable renderer.

Losses. We have formulated a loss function applied jointly to under-controlled coloured mesh auto-encoder and in-the-wild coloured mesh decoder, thus enabling supervised and self-supervised end-to-end training. It is formulated as below:

$$\underset{\theta_{\mathbf{E_M}}, \theta_{\mathbf{E_I}}, \theta_{\mathbf{D}}, \mathbf{m}}{\arg\min} \quad L_{rec} + \lambda L_{render}. \tag{4}$$

Where the objective function:

$$\begin{aligned} L_{\text{rec}} = & \sum_i ||\mathcal{D}(E_M(\mathbf{S}_i; \theta_{\mathbf{E_M}}); \theta_{\mathbf{D}}) - \mathbf{S}_i||_2 \\ & + \sum_i ||\mathcal{D}(E_M(\mathbf{A}_i; \theta_{\mathbf{E_M}}); \theta_{\mathbf{D}}) - \mathbf{A}_i||_1 \end{aligned} \tag{5}$$

is applied to enforce shape and texture reconstruction of the coloured mesh auto-encoder, in which l_2 and l_1 norms are applied on shape S and texture A, respectively. The term:

$$L_{render} = \sum_i ||\mathcal{P}(\mathcal{D}(E_I(\mathbf{I}_i; \theta_{\mathbf{E_I}}); \theta_{\mathbf{D}}); \mathbf{m}) - \mathbf{I}_i||_1 \tag{6}$$

represents the pixel-wise reconstruction error for in-the-wild images when applying a mask to only visible facial pixels. We use $\lambda = 0.01$ and gradually increase to 1.0 during training.

4. Experimental Results

4.1. Datasets

We train our method using both under-controlled data (3DMD [13]) and in-the-wild data (300W-LP [40] and CelebA [24]). The 3DMD dataset [13] contains around 21k raw scans of 3,564 unique identities with expression variations. The 300W-LP dataset [40] consists of about 60k large pose facial data, which are synthetically generated by the profiling method of [40]. The CelebA dataset [24] is a large-scale face attributes dataset with more than 200k celebrity images, which cover large pose variations and background clutter. Each training image is cropped to bounding boxes of indexed 68 facial landmarks with random perturbation to simulate a coarse face detector.

We perform extensive qualitative experiments on AFLW2000-3D [40], 300VW [30] and CelebA testset [24]. We also conducted quantitative comparisons with prior works on FaceWarehouse [8] and Florence [1], where accurate 3D meshes are available for evaluation. FaceWarehouse is a 3D facial expressions database collected by a Kinect RGBD camera. 150 candidates aged from 7 to 80 of various ethnic groups are involved. Florence is a 3D face dataset that contains 53 subjects with their ground truth 3D meshes acquired from a structured-light scanning system.

4.2. Implementation Details

Network Architecture. Our architecture consists of four sub-modules as shown in Fig. 2, named Image Encoder [37, 38], Coloured Mesh Encoder [26], a shared Coloured Mesh Decoder [26] and a differentiable rendering module [17]. The image encoder part takes input images of shape $112 \times 112 \times 3$ followed by 10 convolution layers. It reduces the dimension of the input images to $7 \times 7 \times 256$ and applies a fully connected layer that constructs a 256×1-dimension embedding space. Every convolutional layer is followed by a batch normalization layer and a ReLU activation layer. The kernel size of all convolution layers is 3 and the stride is 2 for any down-sampling convolution layer. The coloured mesh decoder takes an embedding of size 256×1 and decodes to a coloured mesh of size 28431×6 (3 shape and 3 texture channels). The encoder/decoder consists of 4 geometric convolutional filters [26], each one of which is followed by a down/up-sampling layer that reduces/increases the number of vertices by 4 times. Every graph convolutional layer is followed by a ReLU activation function similar to those in the image encoder.

Training Details. Both (1) the under-controlled coloured mesh auto-encoder and (2) the in-the-wild coloured mesh decoder are jointly trained end-to-end although each one uses a different data source. Both models are trained with Adam optimizer with a start learning rate of 1e-4. A learning rate decay is applied with the rate at 0.98 of each epoch. We train the model for 200 epochs. We perturb the training image with a random flipping, random rotation, random scaling and random cropping to the size of 112×112 from a 136×136 input.

4.3. Ablation Study on Coloured Mesh Auto-Encoder

Reconstruction Capacity. We compare the power of linear and non-linear 3DMMs in representing real-world 3D scans with different embedding dimensions to emphasize the compactness of our coloured mesh decoder. Here, we use 10% of 3D face scans from the 3DMD dataset as the test set.

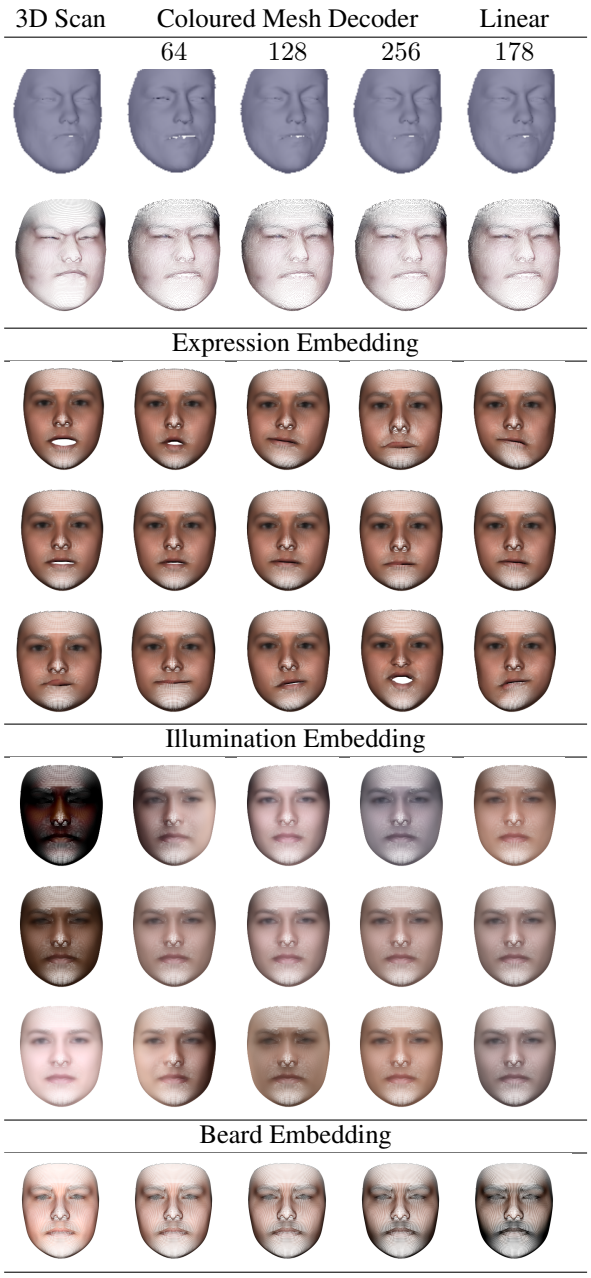


Figure 3. Shape and texture representations followed by expression, illumination and beard embedding generated by the proposed coloured mesh decoder.

Dimension	Shape	Texture
PCA $f_{S/A}$=64	0.0313	0.0196
PCA $f_{S/A}$=128	0.0280	0.0169
PCA $f_{S/A}$=185	0.0237	0.0146
f_S=64	0.0304	-
f_S=128	0.0261	-
f_S=256	0.0199	-
f_{SA}=64	0.0286	0.0325
f_{SA}=128	0.0220	0.0271
f_{SA}=256	**0.0133**	0.0228

Table 1. 3D scan face reconstructions comparison (NME for shape and l_1 channel-wise error for texture).

As illustrated in the top of Fig. 3, we compare the visual quality of reconstruction results produced by linear and non-linear models. To quantify the results of shape modelling, we use the Normalized Mean Error (NME), which is the averaged per-vertex errors between the ground-truth shapes and the reconstructed shapes normalized by inter-ocular distances. For evaluation of texture modelling, we employ the pixel-wise Mean Absolute Error (MAE) between the ground-truth and reconstructed texture.

As shown in Tab. 1, our non-linear shape model has a significantly smaller shape reconstruction error than the linear model. Moreover, the joint non-linear model notably reduces the reconstruction error even further, indicating that integrating texture information is helpful to constrain the deformation of vertices. For the comparison on the texture reconstruction, a slightly higher reconstruction error of texture is expected as the missing texture information between vertices was interpolated in our model, while a linear model has the full texture information.

Attribute Embedding. To get a better understanding of different faces embedded in our coloured mesh decoder, we investigate the semantic attribute embedding. For a given attribute, *e.g.*, smile, we feed the face data (shape and texture) with that attribute $\{\mathbf{I}_i\}_{i=1}^n$ into our coloured mesh encoder to obtain the embedding parameters $\{\mathbf{f}_{SA}^i\}_{i=1}^n$, which represent corresponding distributions of the attribute in the low dimensional embedding space. Taking the mean parameters $\bar{\mathbf{f}}_{SA}$ as input to the trained coloured mesh decoder, we can reconstruct the mean shape and texture with that attribute. Based on the principal component analysis on the embedding parameters $\{\mathbf{f}_{SA}^i\}_{i=1}^n$, we can conveniently use one variable (principal component) to change the attribute. Fig. 3 shows some 3D shapes with texture sampled from the latent space. Here, we can observe that the power of our non-linear coloured mesh decoder is excellent at modelling expressions, illuminations and even beards with a tight embedding dimension ($f_{SA} = 256$).

Method	3DDFA[40]	N3DMM [38]	PRNet [14]	CMD
NME	5.42	4.12	3.62	3.98

Table 2. Face alignment results (%) on the AFLW2000-3D dataset. Performance is reported as bounding box size normalized mean error [40].

Figure 4. Face alignment results on the AFLW2000-3D dataset. The proposed method can handle extreme pose, expression, occlusion and illumination.

4.4. Coloured Mesh Decoder Applied In-the-wild

4.4.1 3D Face Alignment

Since our method can model shape and texture simultaneously, we apply it for 3D morphable fitting in the wild and test the performance on the task of sparse 3D face alignment. We compare our model with the most recent state-of-the-art methods, *e.g*. 3DDFA [40], N-3DMM [37] and PRNet [14] on the AFLW2000-3D [40] dataset. The accuracy is evaluated by the Normalized Mean Error (NME), that is the average of landmark error normalized by the bounding box size on three pose subsets [40].

3DDFA [40] is a cascade of CNNs that iteratively refines its estimation in multiple steps. N-3DMM [38] utilizes the 2D deep convolutional neural networks to build a non-linear 3DMM on the UV position and texture maps, and fits the unconstrained 2D in-the-wild face images in a weakly supervised way. By contrast, our method employs the coloured mesh decoder to build the non-linear 3DMM. Our model not only has better performance but also has a more compact model size and a more efficient running time. PRNet [38] employs an encoder-decoder neural network to directly regress the UV position map. The performance of our method is slightly worse than PRNet majorly due to the complexity of the network.

In Fig. 4, we give some exemplary alignment results, which demonstrate successful sparse 3D face alignment results under extreme poses, exaggerated expressions, heavy occlusions and variable illuminations. We also see that the dense shape (vertices) predictions are also very robust in the wild, which means that for any kind of facial landmark configuration our method is able to give accurate localization results if the landmark correspondence with our shape configuration is given.

4.4.2 3D Face Reconstruction

We first qualitatively compare our approach with five recent state-of-the-art 3D face reconstruction methods: (1) 3DMM fitting networks learned in a supervised way (Sela *et al*. [31]), (2) 3DMM fitting networks learned in an unsupervised way named MoFA (Tewari *et al*. [35]), (3) a direct volumetric CNN regression approach called VRN (Jackson *et al*. [19]), (4) a direct UV position map regression method named PRNet (Feng *et al*. [14]), (5) a non-linear 3DMM fitting networks learned in weakly supervised fashion named N-3DMM (Tran *et al*. [38]). As PRNet and N-3DMM both employ 2D convolution networks on the UV position map to learn the shape model, we view PRNet and N-3DMM as the closest baselines to our method.

Comparison to Sela *et al*. [31]. Their elementary image-to-image network is trained on synthetic data generated by the linear model. Due to the domain gap between synthetic and real images, the network output tends to be unstable on some occluded regions for the in-the-wild testing (Fig. 5), which leads to failure in later steps. By contrast, our coloured mash decoder is trained on the real-world unconstrained dataset in an end-to-end self-supervised fashion, thus our model is robust in handling the in-the-wild variations. In addition, the method of Sela *et al*. [31] requires a slow off-line nonrigid registration step ($\sim$ 180s) to obtain a hole-free reconstruction from the predicted depth map. Nevertheless, the proposed coloured mesh decoder can run extremely fast. Furthermore, our method is complementary to Sela *et al*. [31]'s fine detail reconstruction module. Employing Shape from Shading (SFS) [20] to refine our fitting results could lead to better results with details.

Comparison to MoFA [35]. The monocular 3D face reconstruction method, MoFA, proposed by Tewari *et al*. [35], employs an unsupervised fashion to learn 3DMM fitting in the wild. However, their reconstruction space is still limited to the linear bases. Hence, their reconstructions suffer from unnatural surface deformations when dealing with very challenging texture,*i.e*. beard, as shown in Fig. 6. By contrast, our method employs a non-linear coloured mesh decoder to jointly reconstruct shape and texture. Therefore, our method can achieve high-quality reconstruction results even under hairy texture.

Comparison to VRN [19]. We also compare our approach with a direct volumetric regression method proposed by Jackson *et al*. [19]. VRN directly regresses a 3D shape volume via an encoder-decoder network with skip connection (*i.e*. Hourglass structure) to avoid explicitly using a linear 3DMM prior. This strategy potentially helps the network to explore a larger solution space than the linear model. How-

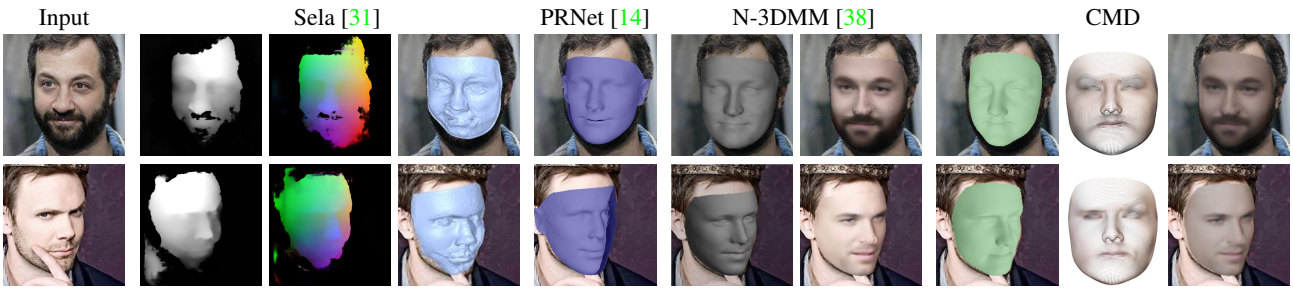


Figure 5. 3D reconstruction results compared to Sela *et al*. [31]. We show the estimated depth, correspondence map and shape for the method proposed by Sela *et al*. [31], and we find occlusions can cause serious problems in their output maps.

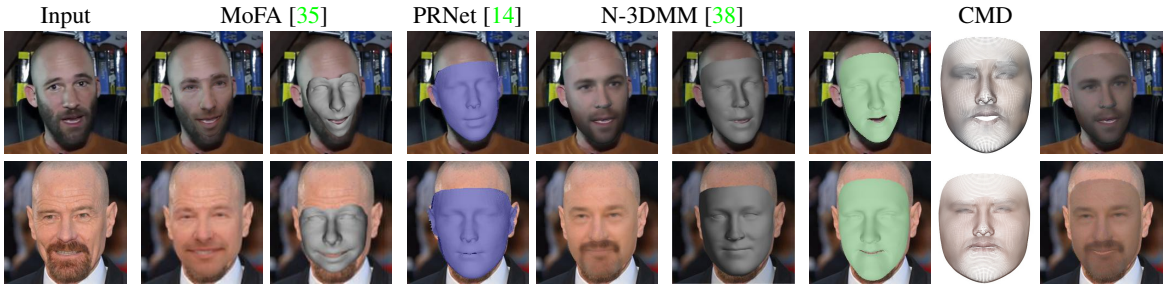


Figure 6. 3D face reconstruction results compared to MoFA [35] on samples from the 300VW dataset [32] (first row) and the CelebA dataset [24] (second row). The reconstructed shapes of MoFA suffer from unnatural surface deformations when dealing with challenging texture, *i.e*. beard. By contrast, our non-linear coloured mesh decoder is more robust to these variations.

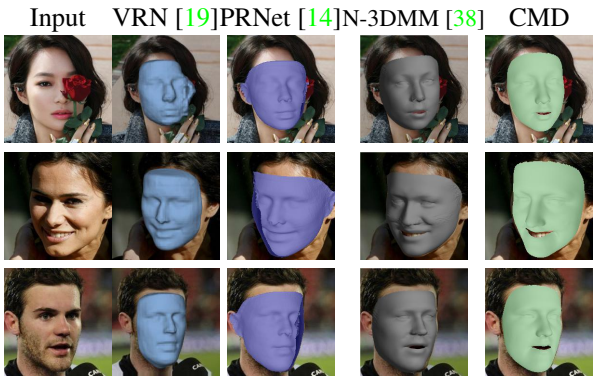


Figure 7. 3D reconstruction results compared to VRN [19] on the CelebA dataset [24]. Volumetric shape representation results in non-smooth 3D shape and loses correspondence between reconstructed shapes. UV position map representation used in PRNet [14] and N-3DMM [38] has comparable performance with our method but the computation complexity is much higher and the model size is much larger.

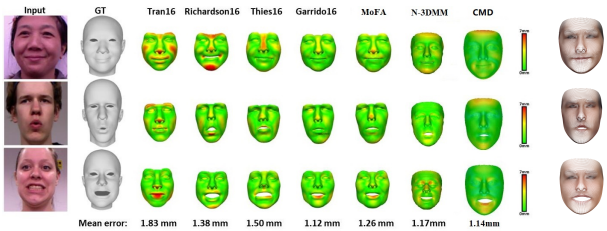


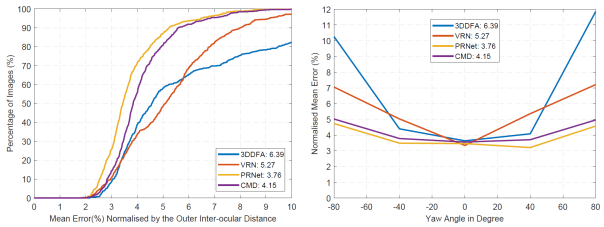
Figure 8. Quantitative evaluation of 3D face reconstruction. We achieved comparable performance compared to Garrido *et al*. [16] and N-3DMM [38].

ever, this method discards the correspondence between facial meshes and the regression target is very large in size. Fig. 7 shows a visual comparison of 3D face reconstructions between VRN and our method. In general, VRN can robustly handle in-the-wild texture variations. However, due to the volumetric shape representation, the surface is not smooth and does not preserve details. By contrast, our method directly models shape and texture of vertices, thus the model size is more compact and the output results are more smooth.

Besides qualitative comparisons with state-of-the-art 3D face reconstruction methods, we also conducted quantitative comparisons on the FaceWarehouse dataset [8] and the Florence dataset [1] to show the superiority of the proposed coloured mesh decoder.

FaceWarehouse. Following the same setting in [35, 38], we also quantitatively compared our method with prior works on 9 subjects from the FaceWarehouse dataset [8]. Visual and quantitative comparisons are illustrated in Fig. 8. We achieved comparable results with Garrido *et al*. [16] and N-3DMM [38], while surpassing all other regression methods [36, 28, 35]. As shown on the right side of Fig. 8, we

can easily infer the expression of these three samples from their coloured vertices.



(a) CED Curves (b) Pose-specific NME

Figure 9. 3D face reconstruction results on the Florence dataset [1]. The Normalized Mean Error of each method is showed in the legend.

Florence. Following the same setting in [19, 14], we also quantitatively compared our approach with state-of-the-art methods (*e.g.* VRN [19] and PRNet [14]) on the Florence dataset [1]. The face bounding boxes were calculated from the ground truth point cloud and the face images were cropped and used as the network input. Each subject was rendered with different poses as in [19, 14]: pitch rotations of $-15°$, $20°$ and $25°$ and raw rotations between $-80°$ and $80°$. We only chose the common face region to compare the performance. For evaluation, we first used the Iterative Closest Points (ICP) algorithm to find the corresponding nearest points between our model output and ground truth point cloud and then calculated Mean Squared Error (MSE) normalized by the inter-ocular distance of 3D coordinates.

Fig. 9(a) shows that our method obtained comparable results with PRNet. To better evaluate the reconstruction performance of our method across different poses, we calculated the NME under different yaw angles. As shown in Fig. 9(b), all the methods obtain good performance under the near frontal view. However, 3DDFA and VRN fail to keep low error as the yaw angle increases. The performance of our method is relatively stable under pose variations and comparable with the performance of PRNet under profile views.

4.5. Running Time and Model Size Comparisons

In Tab. 3, we compare the running time and the model size for multiple 3D reconstruction approaches. Since some methods were not publicly available [31, 35, 38], we only provide an approximate estimation for them. Sela *et al.* [31], VRN [19] and PRNet [14] all use an encoder-decoder network with similar running time. However, Sela *et al.* [31] requires an expensive nonrigid registration step as well as a refinement module.

Our method gets a comparable encoder running time with N-3DMM [38] and MoFA [35]. However, N-3DMM [38] requires decoding features via two CNNs for shape and texture, respectively. MoFA [35] directly uses

	Time		Size	
Method	E	D	E	D
Sela *et al.* [31]	10 ms		1.2G	
VRN [19]	10 ms		1.5G	
PRNet [14]	10 ms		153M	
MoFA [35]	4ms	1.5ms	100M	120M
N-3DMM [38]	2.7ms	5.5 ms	76M	76M
PCA Shape	1.5ms	1.5ms	129M	
PCA Texture	1.7ms	1.7ms	148M	
CMD (f_{SA}=256)	2.7ms	**0.367**ms	76M	**17**M

Table 3. Running time and model size comparisons of various 3D face reconstruction methods. Our coloured mesh decoder can run at 0.367ms on CPU with a compact model size of 17MB.

liner bases, and the decoding step is a single multiplication around 1.5ms for 28K points. By contrast, the proposed coloured mesh decoder only needs one efficient mesh convolution network. On CPU (Intel i9-7900X@3.30GHz), our method can complete coloured mesh decoding within 0.367 ms (2500FPS), which is even faster than using linear shape bases. The model size of our non-linear coloured mesh decoder (17M) is almost one-seventh of the liner shape bases (120MB) employed in MoFA. Most importantly, the capacity of our non-linear mesh decoder is much higher than that of the linear bases as proved in the above experiments.

5. Conclusions

In this paper, we presented a novel non-linear 3DMM method using mesh convolutions. Our method decodes both shape and texture directly on the mesh domain with compact model size (17MB) and very low computational complexity (over 2500 FPS on CPU). Based on the mesh decoder, we propose an image encoder plus a coloured mesh decoder structure that reconstruct the texture and shape directly from an in-the-wild 2D facial image. Extensive qualitative visualization and quantitative reconstruction results confirm the effectiveness of the proposed method.

6. Acknowledgements

Stefanos Zafeiriou acknowledges support from EPSRC Fellowship DEFORM (EP/S010203/1) and a Google Faculty Fellowship. Jiankang Deng acknowledges insightful advice from friends (*e.g.* Sarah Parisot, Yao Feng, Luan Tran and Grigorios Chrysos), financial support from the Imperial President's PhD Scholarship, and GPU donations from NVIDIA.

References

[1] Andrew D Bagdanov, Alberto Del Bimbo, and Iacopo Masi. The florence 2d/3d hybrid face dataset. In *ACM workshop*

on Human gesture and behavior understanding, 2011. 4, 7, 8

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999. 1, 2

[3] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *TPAMI*, 2003. 2

[4] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models in-the-wild. In *CVPR*, 2017. 1, 3

[5] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *IJCV*, 2018. 1

[6] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *CVPR*, 2016. 1, 2

[7] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *SPM*, 2017. 2

[8] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 2014. 4, 7

[9] Shiyang Cheng, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 4dfab: a large scale 4d facial expression database for biometric applications. In *CVPR*, 2018. 2

[10] Grigorios G Chrysos, Jean Kossaifi, and Stefanos Zafeiriou. Robust conditional generative adversarial networks. *ICLR*, 2019. 3

[11] Hang Dai, Nick Pears, William Smith, and Christian Duncan. A 3d morphable model of craniofacial shape and texture variation. In *ICCV*, 2017. 2

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeuriPS*, 2016. 2, 3

[13] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, 2018. 4

[14] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 6, 7, 8

[15] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *CGIT*, 1997. 3

[16] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *TOG*, 2016. 7

[17] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *CVPR*, 2018. 1, 4

[18] Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, 2017. 2

[19] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *ICCV*, 2017. 6, 7, 8

[20] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *TPAMI*, 2011. 6

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2

[22] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *TOG*, 2017. 2

[23] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *CVPR*, 2018. 2

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 4, 7

[25] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 1975. 4

[26] Anurag Ranjan, Timo Bolkart, and Michael J Black. Convolutional mesh autoencoders for 3d face representation. In *ECCV*, 2018. 2, 3, 4

[27] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *ECCV*, 2018. 2

[28] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *CVPR*, 2017. 7

[29] Sami Romdhani and Thomas Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *ICCV*, 2003. 2

[30] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR workshops*, 2013. 4

[31] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, 2017. 1, 6, 7, 8

[32] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *ICCV Workshops*, 2015. 7

[33] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *SPM*, 2013. 2

[34] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *CVPR*, 2018. 1, 2

[35] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, 2017. 6, 7, 8

[36] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *CVPR*, 2017. 1, 2, 7

[37] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *CVPR*, 2018. 1, 2, 4, 6

[38] Luan Tran and Xiaoming Liu. On learning 3d face morphable model from in-the-wild images. *TPAMI*, 2019. 1, 2, 4, 6, 7, 8

[39] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR*, 2018. 2

[40] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016. 2, 4, 6